UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/671,202	09/25/2003	Peter Szor	SYMC1038	6939

34350 7590 03/09/2007

GUNNISON, MCKAY & HODGSON, L.L.P.
1900 GARDEN ROAD, SUITE 220
MONTEREY, CA 93940

EXAMINER	
BAUM, RONALD	
ART UNIT	PAPER NUMBER
2136	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	03/09/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)	
	10/671,202	SZOR, PETER	
	Examiner	Art Unit	
	Ronald Baum	2136	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 4/6/2005.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-24 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-24 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on 27 May 2004 is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date 20070301.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in reply to applicant's correspondence of 06 April 2005.

2. Claims 1-24 are pending for examination.

3. Claims 1-24 are rejected.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

5. Claims 1-4, 7, 8, 14, 18 and 20 are rejected under 35 U.S.C. 101 because the disclosed invention is inoperative and therefore lacks utility. The "determining whether said call ..." per se does not produce a tangible result. For the sake of applying art, the examiner assumes that the method determination aspects are subsequently embodied in a tangible result.

6. Claim 24 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The phrase "A computer program product comprising" is not necessarily embodied software on computer readable media (subject to inclusion of said subject matter in the specification) corresponding to a method of said embodied software. For the sake of applying art, the examiner assumes that the embodied software of the method is so embodied on computer readable media.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

7. Claims 1-24 are rejected under 35 U.S.C. 102(b) as being anticipated by Baratloo, A., et al, 'Transparent Run-Time Defense Against Stack Smashing Attacks', 2000 Proceedings of the USENIX Annual Technical Conference, entire document, http://citeseer.ist.psu.edu/cache/papers/cs/24655/http:zSzzSzwww.research.avayalabs.comzSzprojectzSzlibsafezSzdoczSzusenix00.pdf/baratloo00transparent.pdf ('Baratloo').

8. As per claim 1; "A method comprising:

stalling a call to

a critical operating system (OS) function [Sections 4-7 generally, and more particularly section 6, whereas the libverify 'return address verification scheme ...' techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures, encompasses the claimed limitations as broadly interpreted by the examiner.]; and

determining whether said call is from

a return instruction [Sections 4-7 generally, and more particularly section 6, whereas the libverify 'return address verification scheme ...' techniques, encompasses the claimed limitations as broadly interpreted by the examiner.].".

9. Claim 2 ***additionally recites*** the limitation that; "The method of Claim 1 wherein

said determining whether said call is from a return instruction comprises:

looking up a value at

a previous top of stack; and\

determining whether said value is

equivalent to an address of

said critical OS function.".

The teachings of Baratloo are directed towards such limitations (i.e., Sections 4-8 generally, and more particularly sections 6-8, whereas the libverify 'return address verification scheme ...' techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures such that the Intel IA32 Processors process stack architecture is used, and therefore clearly encompasses the claimed limitations as broadly interpreted by the examiner.).

10. Claim 3 ***additionally recites*** the limitation that; "The method of Claim 2 wherein

a determination is made that said call

is from a return instruction when

a determination is made that said value is

equivalent to said address of

said critical OS function.".

The teachings of Baratloo are directed towards such limitations (i.e., Sections 4-7 generally, and more particularly section 6, whereas the libverify 'return address verification scheme ...'

techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures, encompasses the claimed limitations as broadly interpreted by the examiner.).

11. Claim 4 ***additionally recites*** the limitation that; "The method of Claim 2 wherein

a determination is made that said call

is not from a return instruction when

a determination is made that said value is

not equivalent to said address of

said critical OS function.".

The teachings of Baratloo are directed towards such limitations (i.e., Sections 4-7 generally, and more particularly section 6, whereas the libverify 'return address verification scheme …' techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures, encompasses the claimed limitations as broadly interpreted by the examiner.).

12. Claim 5 ***additionally recites*** the limitation that; "The method of Claim 2 further comprising

taking protective action to protect a computer system upon

a determination that said value is

equivalent to said address of

said critical OS function.".

The teachings of Baratloo are directed towards such limitations (i.e., Sections 4-7 generally, and more particularly section 6, whereas the libverify 'return address verification scheme ...' techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures such that upon determination of an attack/non-attack scenario, the `die()` function is called/not called (and associated subsequent logging as a `syslog` entry either way), encompasses the claimed limitations as broadly interpreted by the examiner.).

13. Claim 6 ***additionally recites*** the limitation that; "The method of Claim 2 further comprising

allowing said call to proceed upon

a determination that said value is

not equivalent to said address of

said critical OS function.".

The teachings of Baratloo are directed towards such limitations (i.e., Sections 4-7 generally, and more particularly section 6, whereas the libverify 'return address verification scheme ...' techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures such that upon determination of an attack/non-attack scenario, the

`die()` function is called/not called (and associated subsequent logging as a `syslog` entry either way), encompasses the claimed limitations as broadly interpreted by the examiner.).

14. Claim 7 ***additionally recites*** the limitation that; "The method of Claim 2 wherein said previous top of stack is

at address [ESP-4].".

The teachings of Baratloo are directed towards such limitations (i.e., Sections 4-7 generally, and more particularly section 6, whereas the libverify 'return address verification scheme ...' techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures such that the Intel IA32 Processors process stack architecture is used, and therefore clearly encompasses the claimed limitations as broadly interpreted by the examiner.).

15. Claim 8 ***additionally recites*** the limitation that; "The method of Claim 7 wherein a top of stack is

at address [ESP].".

The teachings of Baratloo are directed towards such limitations (i.e., Sections 4-7 generally, and more particularly section 6, whereas the libverify 'return address verification scheme ...' techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures such that the Intel IA32 Processors process stack architecture is

used, and therefore clearly encompasses the claimed limitations as broadly interpreted by the examiner.).

16. Claim 9 ***additionally recites*** the limitation that; "The method of Claim 1 wherein

upon a determination that

said call is from

said return instruction during said determining,

said method further comprising

taking protective action to protect a computer system.".

The teachings of Baratloo are directed towards such limitations (i.e., Sections 4-7 generally, and more particularly section 6, whereas the libverify 'return address verification scheme …' techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures such that upon determination of an attack/non-attack scenario, the `die()` function is called/not called (and associated subsequent logging as a `syslog` entry either way), encompasses the claimed limitations as broadly interpreted by the examiner.).

17. Claim 10 ***additionally recites*** the limitation that; "The method of Claim 9 wherein

said taking protective action comprises

terminating said call.".

The teachings of Baratloo are directed towards such limitations (i.e., Sections 4-8 generally, and more particularly sections 6-8, whereas the libverify 'return address verification scheme …'

techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures such that upon determination of an attack/non-attack scenario, the `die()` function is called/not called (and associated subsequent logging as a `syslog` entry either way), encompasses the claimed limitations as broadly interpreted by the examiner.).

18. Claim 11 ***additionally recites*** the limitation that; "The method of Claim 9 wherein said taking protective action comprises

terminating a call module originating said call.".

The teachings of Baratloo are directed towards such limitations (i.e., Sections 4-8 generally, and more particularly sections 6-8, whereas the libverify 'return address verification scheme ...' techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures such that upon determination of an attack/non-attack scenario, the `die()` function is called/not called (and associated subsequent logging as a `syslog` entry either way), encompasses the claimed limitations as broadly interpreted by the examiner.).

19. Claim 12 ***additionally recites*** the limitation that; "The method of Claim 9 wherein said taking protective action comprises

terminating a parent application comprising

a call module originating said call.".

The teachings of Baratloo are directed towards such limitations (i.e., Sections 4-8 generally, and more particularly sections 6-8, whereas the libverify 'return address verification scheme ...' techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures such that upon determination of an attack/non-attack scenario, the `die()` function is called/not called (and associated subsequent logging as a `syslog` entry either way), encompasses the claimed limitations as broadly interpreted by the examiner.).

20. Claim 13 ***additionally recites*** the limitation that; "The method of Claim 9 further comprising

providing a notification that

said protective action has been taken.".

The teachings of Baratloo are directed towards such limitations (i.e., Sections 4-8 generally, and more particularly sections 6-8, whereas the libverify 'return address verification scheme ...' techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures such that upon determination of an attack/non-attack scenario, the `die()` function is called/not called (and associated subsequent logging as a `syslog` entry either way), encompasses the claimed limitations as broadly interpreted by the examiner.).

21. Claim 14 ***additionally recites*** the limitation that; "The method of Claim 1 wherein

upon a determination that

said call is from

said return instruction during said determining,

said method further comprising

determining whether said call is

a known false positive.".

The teachings of Baratloo are directed towards such limitations (i.e., Sections 4-8 generally, and more particularly sections 6-8, whereas the libverify 'return address verification scheme ...' techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures such that upon determination of an attack/non-attack scenario, the `die()` function is called/not called (and associated subsequent logging as a `syslog` entry either way) such that attack scenarios both known and *previously unknown*, encompasses the claimed limitations as broadly interpreted by the examiner.).

22. Claim 15 ***additionally recites*** the limitation that; "The method of Claim 14 wherein upon a determination that

said call is

not said known false positive,

said method further comprising

taking protective action to protect a computer system.".

The teachings of Baratloo are directed towards such limitations (i.e., Sections 4-8 generally, and more particularly sections 6-8, whereas the libverify 'return address verification scheme ...'

techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures such that upon determination of an attack/non-attack scenario, the `die()` function is called/not called (and associated subsequent logging as a `syslog` entry either way) such that attack scenarios both known and *previously unknown*, encompasses the claimed limitations as broadly interpreted by the examiner.).

23. Claim 16 ***additionally recites*** the limitation that; "The method of Claim 15 further comprising

providing a notification that

said protective action has been taken.".

The teachings of Baratloo are directed towards such limitations (i.e., Sections 4-8 generally, and more particularly sections 6-8, whereas the libverify 'return address verification scheme …' techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures such that upon determination of an attack/non-attack scenario, the `die()` function is called/not called (and associated subsequent logging as a `syslog` entry either way) such that attack scenarios both known and *previously unknown*, encompasses the claimed limitations as broadly interpreted by the examiner.).

24. Claim 17 ***additionally recites*** the limitation that; "The method of Claim 14 wherein

upon a determination that

said call is

said known false positive,

said method further comprising

allowing said call to proceed.".

The teachings of Baratloo are directed towards such limitations (i.e., Sections 4-8 generally, and more particularly sections 6-8, whereas the libverify 'return address verification scheme …' techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures such that upon determination of an attack/non-attack scenario, the `die()` function is called/not called (and associated subsequent logging as a `syslog` entry either way) such that attack scenarios both known and *previously unknown*, encompasses the claimed limitations as broadly interpreted by the examiner.).

25. Claim 18 ***additionally recites*** the limitation that; "The method of Claim 1 further comprising

hooking

said critical OS function.".

The teachings of Baratloo are directed towards such limitations (i.e., Sections 4-7 generally, and more particularly section 6, whereas the libverify 'return address verification scheme …' techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel (i.e., hooking), and associated modification and retention of user

code size, location, addressing structures, encompasses the claimed limitations as broadly interpreted by the examiner.).

26. Claim 19 ***additionally recites*** the limitation that; "The method of Claim 1 further comprising

originating said call to

said critical OS function.".

The teachings of Baratloo are directed towards such limitations (i.e., Sections 4-8 generally, and more particularly sections 6-8, whereas the libverify 'return address verification scheme ...' techniques with the associated system call/return (i.e., critical OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures such that upon determination of an attack/non-attack scenario, the `die()` function is called/not called (and associated subsequent logging as a `syslog` entry either way) such that attack scenarios both known and *previously unknown*, encompasses the claimed limitations as broadly interpreted by the examiner.).

27. Claim 20 ***additionally recites*** the limitation that; "The method of Claim 1 wherein said critical OS function is necessary for

a first application to cause

execution of a second application.".

The teachings of Baratloo are directed towards such limitations (i.e., Sections 4-8 generally, and more particularly sections 6-8, whereas the libverify 'return address verification scheme ...'

techniques with the associated system call/return (i.e., critical OS call and associated chained sequences of calling via appropriate call parameter passing stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures such that upon determination of an attack/non-attack scenario, the `die()` function is called/not called (and associated subsequent logging as a `syslog` entry either way) such that attack scenarios both known and *previously unknown*, encompasses the claimed limitations as broadly interpreted by the examiner.).

28. Claim 21 ***additionally recites*** the limitation that; "The method of Claim 20 wherein said second application allows

remote access of a computer system.".

The teachings of Baratloo are directed towards such limitations (i.e., Sections 4-8 generally, and more particularly sections 6-8, whereas the libverify 'return address verification scheme ...' techniques with the associated system call/return (i.e., critical OS call and associated chained sequences of calling via appropriate call parameter passing, both locally and networked objects access stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures such that upon determination of an attack/non-attack scenario, the `die()` function is called/not called (and associated subsequent logging as a `syslog` entry either way) such that attack scenarios both known and *previously unknown*, encompasses the claimed limitations as broadly interpreted by the examiner.).

29. As per claim 22; "A method comprising:

stalling a call to

a critical operating system (OS) function [Sections 4-7 generally, and more particularly section 6, whereas the libverify 'return address verification scheme ...' techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures, encompasses the claimed limitations as broadly interpreted by the examiner.];

looking up a value at

a previous top of stack [Sections 4-8 generally, and more particularly sections 6-8, whereas the libverify 'return address verification scheme ...' techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures such that the Intel IA32 Processors process stack architecture is used, and therefore clearly encompasses the claimed limitations as broadly interpreted by the examiner.]; and

determining whether said value is

equivalent to an address of

said critical OS function [Sections 4-8 generally, and more particularly sections 6-8, whereas the libverify 'return address verification scheme ...' techniques with the associated system call/return (i.e., critical OS call stalling) interceptor called as part of the operating system kernel, and associated

modification and retention of user code size, location, addressing structures such that upon determination of an attack/non-attack scenario, the `die()` function is called/not called (and associated subsequent logging as a `syslog` entry either way) such that attack scenarios both known and *previously unknown*, encompasses the claimed limitations as broadly interpreted by the examiner.],

wherein upon a determination that

said value is equivalent to said address of said critical OS function,

said method further comprising

taking protective action to protect a computer system [Sections 4-8 generally, and more particularly sections 6-8, whereas the libverify 'return address verification scheme ...' techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures such that upon determination of an attack/non-attack scenario, the `die()` function is called/not called (and associated subsequent logging as a `syslog` entry either way), encompasses the claimed limitations as broadly interpreted by the examiner.].".

30. Claim 23 ***additionally recites*** the limitation that; "The method of Claim 22 wherein upon a determination that said value is

not equivalent to said address of said critical OS function,

said method further comprising

allowing said call to proceed.".

The teachings of Baratloo are directed towards such limitations (i.e., Sections 4-7 generally, and more particularly section 6, whereas the libverify 'return address verification scheme ...' techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures such that upon determination of an attack/non-attack scenario, the `die()` function is called/not called (and associated subsequent logging as a `syslog` entry either way), encompasses the claimed limitations as broadly interpreted by the examiner.).

31. As per claim 24; "A computer program product comprising:

a Return-to-LIBC attack blocking application for

stalling a call to

a critical operating system (OS) function [Sections 4-7 generally, and more particularly section 6, whereas the libverify 'return address verification scheme ...' techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures, encompasses the claimed limitations as broadly interpreted by the examiner.];

said Return-to-LIBC attack blocking application further

for looking up a value at

a previous top of stack [Sections 4-8 generally, and more particularly sections 6-8, whereas the libverify 'return address verification scheme ...'

techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures such that the Intel IA32 Processors process stack architecture is used, and therefore clearly encompasses the claimed limitations as broadly interpreted by the examiner.]; and

said Return-to-LIBC attack blocking application further

for determining whether said value is

equivalent to an address of

said critical OS function [Sections 4-8 generally, and more particularly sections 6-8, whereas the libverify 'return address verification scheme ...' techniques with the associated system call/return (i.e., critical OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures such that upon determination of an attack/non-attack scenario, the `die()` function is called/not called (and associated subsequent logging as a `syslog` entry either way) such that attack scenarios both known and *previously unknown*, encompasses the claimed limitations as broadly interpreted by the examiner.],

wherein upon a determination that

said value is equivalent to said address of said critical OS function,

said Return-to-LIBC attack blocking application further

for taking protective action to protect a computer system comprising

said Return-to-LIBC attack blocking application [Sections 4-8 generally, and more particularly sections 6-8, whereas the libverify 'return address verification scheme ...' techniques with the associated system call/return (i.e., OS call stalling) interceptor called as part of the operating system kernel, and associated modification and retention of user code size, location, addressing structures such that upon determination of an attack/non-attack scenario, the `die()` function is called/not called (and associated subsequent logging as a `syslog` entry either way), encompasses the claimed limitations as broadly interpreted by the examiner.].".

Conclusion

32. Any inquiry concerning this communication or earlier communications from examiner should be directed to Ronald Baum, whose telephone number is (571) 272-3861, and whose unofficial Fax number is (571) 273-3861 and unofficial email is Ronald.baum@uspto.gov. The examiner can normally be reached Monday through Thursday from 8:00 AM to 5:30 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Nasser Moazzami, can be reached at (571) 272-4195. The Fax number for the organization where this application is assigned is **571-273-8300**.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. For more information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Ronald Baum

Patent Examiner

NASSER MOAZZAMI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

3,7,07